SoundScape

PRO MIDI STUDIO

ADDENDUM

mimetics™

corporation

...the Professional Software Source!!

Table of Contents

1. Introduction 1
2. Multiple Edit Windows 1
3. Directory Support in the Load Requester 1
4. Default Disk for Loads and Saves 1
5. Beats and Measures 2
5.1 Time Signature 2
5.2 Interpreting the Time 3
6. The Move Command 4
7. Edit Positioning 4
8. Auto Insert Disable 4
9. MIDI Insert Editing 5
10. Undo 6
11. Panic Button 6
12. SMUSConvert 7
12.1 Converting from SoundScape to SMUS 7
12.2 Converting from SMUS to SoundScape 8
13. New Modules File 8
14. SoundScape as a Library 9

Introduction

We've put some new features into SoundScape. These notes cover these features and conclude with a discussion on how SoundScape is implemented as a "library" on the Amiga.

Multiple Edit Windows

The SoundScape Patch Panel now supports opening multiple edit windows. Previously, if an edit window was opened from the Patch Panel, all further Patch Panel activities such as opening another edit window or connecting two modules, were not allowed until that edit window was closed.

Now, you can open an edit window (by clicking twice on an icon) and continue to work with the Patch Panel.

Directory Support in the Load Requester

When you execute a LOAD command (be it environment, samples, tracks, etc.), it finds all the files of the proper type in the selected directory. It now also includes the names of subdirectories (drawers) that exist within this directory. These subdirectories are labeled with "`(dir)`" after their name. To access a subdirectory, click on its name twice. SoundScape will search this subdirectory and display the files found. In addition, the subdirectory name is now appended to the directory name above the file name list. To leave this subdirectory, simply click on the directory name and edit it out.

Default Disk for Loads and Saves

When you first execute a load command, you will notice that, instead of looking for files in the volume "SoundScape:", it now looks in the volume "Default:".

With AmigaDos, you can assign any arbitrary volume name to any directory. So, if AmigaDos has been told beforehand that "Default:" is, infact, "SoundScape", it will automatically look there.

The AmigaDos CLI command to make this assignment is called "`Assign`". So, if you typed

```
Assign Default:  SoundScape:
```

from the CLI, from then on the name "Default:" could be used as a name for the volume "SoundScape:".

If you ran the installation program for your SoundScape (be it an update or the new product), it set up the modules file (see the section "New Modules File") to do the above assignment. So, when you do your first Load and it displays "Default:" as the directory instead of "SoundScape:", it still reads the files from your SoundScape disk.

What does this give you? You can now change your default directory to be something other than "SoundScape:". This comes in real handy if you have a hard disk.

Beats and Measures

With the SoundScape Sequence and Song editors you can now edit time values by beat and measure, instead of just by clock ticks.

Time Signature

In the Edit Sequence window, there is a new string gadget:

```
T Sig 2/4      B
```

By editing this, you decide what time signature the sequence should be displayed in. The number on the left side of the slash (in this case, 2) is the number of beats per measure. This can be any number from 1 to 99. The number on the right hand side of the slash says what type of note is used to define a beat. For example, 4 indicates a quarter note, 8 an eighth note.

A quarter note is 24 MIDI clock ticks.

You might prefer defining the beat not by the type of note but by the duration of the note in MIDI clocks. For example, although a quarter note can be used to represent 24 clocks, there is no note that represents 25 clocks to the beat. Should you be so perverse as to desire your song broken into measures defined by nonstandard notes, click on the 'B'. It turns to 'C', indicating you can now define the beat in clocks. So, 4/4 turns into 4/24. We aim to please.

If you would rather stick with clock values, just set either number to 0.

Interpreting the Time

With beats and measures, time values are displayed like this:

`measure:beat:clocks`

Example:

`10: 1: 12`

This says, "We're ten measures into the sequence, in the first beat of that measure, and we've advanced by 12 clocks in the beat."

Measures and beats always start with 1 (the first beat or measure) and count up. Clocks start from 0. So, `1:1:0` actually is the very start, being the first measure, first beat, with no clocks so far. This might be confusing at first, so play around with it a bit.

When you edit a time value, if you put in numbers that don't make sense, it should reject the edit and display the original value.

There are four places where you actually edit time values. In the Edit Sequence window, there are two:

`Time 10: 1:12`

`Next 10:20:0`

`Time` specifies the moment in time the current MIDI event occurs. Should you edit it, you will reposition the note without altering any of the others in the sequence.

`Next` defines when the note that follows should occur. Changing this will shift all notes that follow by the same amount. Unlike older SoundScapes (and the manual), this is an absolute value. In older SoundScapes, this was a relative value. You specified how long after this event the next event occured. Now this number is simply the actual time of that event.

Should you attempt to change the next time value to a value prior to the current note, it will ignore your attempt.

In the Song window, there is one place you can edit a time value:

`Start 19: 4:0`

This used to be called 'Wait', because it defined how long to wait after the end of the last sequence before starting this sequence. Now it simply defines in absolute terms when the first event of the sequence occurs.

Finally, from the Tape Deck itself, we can edit in an initial delay:

```
Delay   20:  1:0
```

The Move Command

The Edit Song window has an additional feature, the 'Move' command. You can use this to rearrange songs. Move a sequence around within a song by first clicking on it, then clicking 'Move', then clicking on the position in the song list you'd like the sequence to be.

Edit Positioning

If you play, rewind, or fast forward to a particular point in your song, then enter either the Sequence or Song editor, the cursor will automatically go to that current position in the song.

Auto Insert Disable

You can enable and disable the Punch In and Punch Out registers. To enable the Punch In register, click on the 'PI' button to the left of it. The button will turn red, indicating it is enabled. Then, when you set the Tape Deck running, it will enter record mode when the location indicated in the Punch In register is reached. Disable Punch In by clicking again on the 'PI' button.

You can enable and disable the Punch Out register by clicking on the 'PO' button.

MIDI Insert Editing

You can now enter MIDI events into the sequence editor by playing them from a keyboard. This can be the console keyboard, an external MIDI keyboard, or any other MIDI device. Here's how to use it:

- Select the track you wish to edit in the Track List.
- Set the track in RECORD mode. This enables insert editing.
- Click on "ST IN" to edit which types of events you wish to insert. Generally, you would only be interested in Note events, so disable all others, such as Pitch Wheel.
- Make sure the device you wish to enter notes with is set up as the module in for this track, with the proper channel in set.
- Click on "Sequence" to enter the Sequence Editor.
- Select where you wish to edit. Do this by selecting a note in the note list. The inserted note will be put there (without deleting the selected one.)
- Play a note. It will be inserted and displayed.

If you append a note at the end of the list, it will be placed after the previous last note by the amount defined in the "Next" field of the last note. In this way, you can quickly enter notes in step time. Try it. It's great for designing arpeggios for Echo mode, among other things.

If the note is entered somewhere in the middle of the list, it doesn't adjust the timing. Instead, it has this note occur at the same time as the selected one, creating a chord. You can now edit it if you wish to put it somewhere else.

Undo

You are now given the ability to enable and disable the Undo command. In order for Undo to work, copies have to be made of all sequences prior to recording. This eats up a lot of memory, especially if you're recording lots of pitchbend and aftertouch. So if you don't plan to use Undo, it's often wise to disable it.

To alternately enable and disable Undo, click on the new icon placed just to the right of the Undo button.

By default, Undo is disabled.

Panic Button

It is possible to gridlock SoundScape.

For example, you could have a whole bunch of modules hooked up with a feedback loop somewhere in there. A note would enter this feedback loop and cycle infinitely, resulting in the system acting as if it were hung (but it's really just doing its job.)

When this problem occurs, all the windows can be moved and icons clicked, but nothing new happens.

There's a way out. Hidden behind the Patch Panel is a small window with "Panic Button" displayed in red. You can get at this window by simply moving the Patch Panel out of the way or moving it to the back.

Click on the red panic button. It will clear out all currently transmitting events and tell all modules to stop.

There are other ways of hanging the system. You could describe a sequence that loops with no time increment between events. When playing, the Tape Deck would cycle infinitely, outputing the same events over and over again. It's actually hard to trick the Tape Deck into doing this because the Sequence Editor looks out for such loops. But, if you are determined, you can always find a way. The Panic Button will pull the Tape Deck out of such a loop.

Or, you could simply have too much going on. Try to drive SoundScape mad by turning the clock up to full blast and then running it through a rate scaler module that turns it up even higher and have everything running and processing events at once. If enough is going on, the computer can no longer get everything done on time and the system gets farther and farther behind. The Panic Button will halt this insanity.

However, you probably will never need the Panic Button. It's just nice to know it's there.

SMUSConvert

SMUSConvert is a program that converts SoundScape music files into the IFF SMUS file standard and back.

SMUSConvert runs with SoundScape. You can invoke it while SoundScape is running or beforehand. If you invoke SMUSConvert first, it will start SoundScape (see the section on SoundScape as a Library.)

Converting from SoundScape to SMUS

Under "SoundScape→SMUS" is two text gadgets. With "Quarter Note" you define how many MIDI clocks define a quarter note. It defaults to 24, which is the MIDI standard. You will probably rarely change this.

"Quantization" defines what the resolution of the notes should be. With the SoundScape file format, each note can have any random duration. With SMUS, each note is defined within traditional notational constraints. So, a note with a duration of 18 clocks would be a dotted eighth note. Fine. But a note with a duration of 21 clocks would be a dotted eighth tied to a thirty second note. Things get messy and the rests between notes also need to be defined in this way.

So, there's a real need to quantize the time a note occurs and its duration to a reasonable value. The value is defined in clocks. The default is 12 clocks, or an eighth note. With this quantization, the above example would give you an eighth note for the 18 clock duration and a quarter note for the 21 clock duration. If higher resolution is required, set the quantization number appropriately. A value of 6 would result in resolution down to sixteenth notes. To define a resolution, remember there are 96 clocks to a whole note. Simply divide 96 by the note type to get the number of clocks. 96 / 4 is 24 clocks per quarter note. 96 / 16 is 6 clocks per sixteenth note.

If you define a number that doesn't divide into 96, like 13, it will not define any legal note type. The results will probably be strange and useless.

Play around with different resolutions and see what they generate.

Having set the quantization and quarter note values, click on "SoundScape → SMUS". A standard load requester will open, asking you which file to load. This will be loaded and converted to IFF. Then another requester will ask you what the file should be saved as. '.SMUS' will be appended to the file name as it is saved.

If there are tracks in your score that you would rather not have converted into SMUS, such as a drum track, you can disable conversion of it by setting it Mute when saving to disk.

All looped sequences and song tracks are played out until the score's end, which is defined by the longest non-looped sequence or song.

SMUS is not a performance format. Pitch wheel, after touch, etc., are removed.

Converting from SMUS to SoundScape

Under the "SMUS→SoundScape" icon, there is another text gadget that asks you to define how many clocks there should be per quarter note. It defaults to 24, which is the MIDI standard.

If the file does not have ".SMUS" appended to the end, you will need to type in the whole name of the file. The box under "SMUS File Name" lets you do that.

Click on "SMUS→SoundScape". If you didn't specify a file name, a standard load requester will prompt you for one. The file will be loaded and converted to the SoundScape format. You will then be asked to give a new file name and it will be saved.

New Modules File

Along with the modules that come initially with SoundScape, additional ones can be bought and installed so that they become part of the SoundScape system, appearing in the Patch Panel when SoundScape starts running.

In order for these modules to come up as part of SoundScape, SoundScape has to find them and know to install them. This is accomplished by having a file with a list of module commands in it. On start up, SoundScape opens this file and reads the module instructions from it.

Up until now, this file has been `SoundScape:ss/modules`. But this means reading and occasionally updating a file on your master copy of SoundScape, something it would be nice to avoid. So, an alternative modules file can now be used. It is identical in format to `SoundScape:ss/modules` but can exist on your WorkBench disk. Name it `SS-Modules` and place it in the `S` directory of your WorkBench disk.

When looking for modules to load, SoundScape will first search for `S:SS-Modules`. If it can't find it, then it will try `SoundScape:ss/modules`.

The format for a modules file should be:

```
run modulename
run anothermodulename
run yetanothermodulename
```

In other words, it is simply a list of CLI commands to execute.

To create and edit this file, use a text editor such as EMACS or Edit.

For those who aren't too comfortable with the idea of editing a CLI command file, we'll be providing a simple program that lets you select which modules you want to autoload. It will change the modules file for you.

This program will be included with the modules.

SoundScape as a Library

SoundScape is not implemented as a single program that you run (although it may appear that way.) Instead, it is implemented as a "Library" on the Amiga. This gives SoundScape much of its power and flexibility.

But it can also cause some confusion, because things aren't quite what you'd expect. So, here's a brief explanation of what's going on:

You don't run SoundScape directly.

Instead, you run a program that invokes SoundScape. This program asks the Amiga's operating system to start SoundScape running. The Amiga's operating system looks for `soundscape.library` in the `libs` directory of your WorkBench disk. If it is there, it loads it in and starts it running. The Patch Panel opens and the modules are displayed.

Then, the program that asked for SoundScape can now work with SoundScape. For example, it can install its icon in the Patch Panel and start sending and receiving notes. In addition, multiple programs can invoke SoundScape at the same time, all working with SoundScape in one way or another.

The very simplest program that uses SoundScape is, infact, also called "SoundScape" (we'll refer to it in quotes), and it is what you are probably familiar with. But if you were to inspect it, you would notice that it is a very small program. All it does is put up a window (that one in the top left corner) and ask the Amiga to start the real SoundScape running. Then it waits for you to close its window. When you do, it simply tells SoundScape to shut down.

But the real SoundScape program that you see doing all the magic with the Patch Panel, Sampler, Tape Deck, etc, is actually `soundscape.library`.

Another example of a program that invokes SoundScape is SMUSConvert. When SMUSConvert starts running, it asks for SoundScape to be run. Once again, the Amiga's operating system loads in `soundscape.library` and sets it running. Then, SMUSConvert uses it to access music files and convert them.

If you run both "SoundScape" and SMUSConvert, you will notice that SoundScape gets invoked only once, and that you need to shut both down (in any order) before SoundScape will close.

Fortunately, all of this library stuff is transparent to you, the user.

But, because of this transparency, if SoundScape doesn't run, you may scratch your head and wonder why. If that happens, here's a checklist of criteria that must be satisfied in order for SoundScape to run:

- Your WorkBench disk has a copy of `soundscape.library` in its `libs` directory. This is done automatically by the program SSinstall, which you run when you first get SoundScape. To run SSinstall, simply reboot with the SoundScape disk as your WorkBench disk. It will tell you what to do.

- You must also have `mathtrans.library` installed in the `libs` directory. This is also handled by SSinstall.

- The WorkBench disk that has `soundscape.library` is also the WorkBench disk you booted with. If you simply swap disks and then invoke SoundScape, the Amiga will ask for the WorkBench disk you booted with, whether or not it has `soundscape.library` in it. If it doesn't, it won't run SoundScape.

- You have a program, such as "SoundScape" or SMUSConvert that can invoke SoundScape.

- There's enough memory. SoundScape will not run on 256k machines. If you are running on a 512k machine with several other programs going, there might not be enough room to load `soundscape.library`.

One important ramification of all this: When you update your SoundScape to a new version, you should be updating `soundscape.library`, not the program "SoundScape". And, make sure you do the update on all WorkBench disks that you are using for SoundScape. Otherwise, depending on which you booted up with, you may or may not be running with the latest version of the program, even though you installed it on the SoundScape disk.